

SISTEMA INTEGRAL DE CALIFICACIÓN AUTOMATIZADA COMO
ALTERNATIVA PARA OPTIMIZACIÓN Y GESTIÓN DE EVALUACIÓN DE
EXAMEN TIPO TEST EN LA UNIVERSIDAD DEL VALLE

FABIÁN ANDRÉS PAYAN MENESES



UNIVERSIDAD DEL VALLE
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
SANTIAGO DE CALI
2012

SISTEMA INTEGRAL DE CALIFICACIÓN AUTOMATIZADA COMO
ALTERNATIVA PARA OPTIMIZACIÓN Y GESTIÓN DE EVALUACIÓN DE
EXAMEN TIPO TEST EN LA UNIVERSIDAD DEL VALLE

FABIÁN ANDRÉS PAYAN MENESES

Trabajo de Grado
Presentado como requisito para optar por el título de
Ingeniero de Sistemas

Director
Ángel De La Encarnación García Baños, Ph. D.



UNIVERSIDAD DEL VALLE
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
SANTIAGO DE CALI
2012

NOTAS DE ACEPTACIÓN

El presente trabajo de grado fue aprobado en cumplimiento de los requisitos expedidos por la institución para optar el título de:

INGENIERO DE SISTEMAS

Presidente del Jurado

Director

AGRADECIMIENTOS

El autor expresa su agradecimiento a:

Ángel De La Encarnación García Baños, director de la Escuela de Ingeniería de Sistemas y Computación, por dirigirme adecuadamente durante la elaboración del proyecto.

Fredy Javier Mosquera Riascos, Ingeniero de Sistemas de la Universidad del Valle, por la asesoría en la realización del proyecto y metodologías ágiles de desarrollo de software

Universidad del Valle, Profesores, compañeros y demás personas involucradas en este proyecto.

Amigos y familia, un especial agradecimiento a mis hermanos y padres, que siempre creyeron en mí y han estado siempre motivándome para seguir adelante.

Dios, gracias a ti por hacer todo esto posible.

TABLA DE CONTENIDO

1. INTRODUCCIÓN.....	7
2. DESCRIPCIÓN DEL PROBLEMA.....	8
3. JUSTIFICACIÓN DE LA SOLUCIÓN PROPUESTA.....	9
4. OBJETIVOS.....	10
4.1 Objetivo General.....	10
4.2 Objetivos Específicos.....	10
5. ALCANCE DEL PROYECTO.....	10
6. MARCO REFERENCIAL.....	11
6.1 Estado del arte.....	11
6.2. Marco Conceptual.....	12
7. METODOLOGÍA.....	16
7.1 Fase de exploración.....	16
7.2 Fase de análisis o planeación.....	17
7.2.1 Historias de usuarios.....	17
7.2.2 Tecnologías del proyecto.....	17
7.3 Fase de diseño.....	18
7.3.1 Diseño computacional.....	18
7.3.1.1 Modelo de navegación.....	18
7.3.2 Arquitectura del sistema.....	19
7.3.2.1 Vista física.....	20
7.3.2.2 Vista lógica.....	21
7.4 Fase de producción.....	22
7.4.1 Primera iteración.....	22
7.4.2 Segunda iteración.....	23
7.4.2 Tercera iteración.....	24
7.5 Fase de pruebas.....	26
7.5.1 Pruebas del sistema.....	27
9. REFERENCIAS BIBLIOGRÁFICAS.....	31

RESUMEN

Este proyecto propone el análisis, diseño e implementación de una herramienta de software para automatizar el proceso de revisión, evaluación y administración de calificaciones en exámenes tipo test de cualquier asignatura. Teniendo en cuenta que es una herramienta automática, se hace necesario que: i) el docente proporcione algunos parámetros; ii) la evaluación sea digitalizada por medio de un escáner; iii) se realice una comparación con un banco de datos de respuestas. Este proyecto se presenta como una alternativa a los sistemas que actualmente dan soporte a este proceso.

La herramienta utiliza el método de reconocimiento automático de marcas OMR, el cual aumenta al máximo la confiabilidad en la información capturada. Además, optimiza el tiempo de respuesta de la emisión de resultados y la administración de las calificaciones almacenadas, brindándole al docente una solución que reduce drásticamente los costos asociados al proceso manual de revisión y evaluación de exámenes.

1. INTRODUCCIÓN

Uno de los procedimientos más habituales para la evaluación de conocimientos de los alumnos es la realización de cuestionarios tipo test con preguntas de opción múltiple con única o múltiple respuesta. Este tipo de pruebas tiene la ventaja de que su corrección es totalmente objetiva; se debe comprobar si el alumno ha respondido la opción correcta o no. De igual forma, otra ventaja es que la corrección puede realizarse rápidamente y esto crea un beneficio adicional en el caso en que el número de evaluaciones es elevado. Por estos motivos, dichos cuestionarios son utilizados cada vez más en todo tipo de evaluaciones.

A lo largo del documento, se plantea una alternativa para agilizar aún más el proceso de calificaciones de los exámenes empleando una técnica en la cual deben ser realizados en papel y utilizando un escáner adecuado junto con su software específico que realiza la corrección.

2. DESCRIPCIÓN DEL PROBLEMA

En la Universidad Del Valle, se llevan a cabo cientos de parciales durante un semestre. En estos parciales se pueden encontrar diversos estilos de preguntas, particularmente las preguntas con opción múltiple (POM) y única respuesta o múltiple respuesta. Al realizar evaluaciones que contengan este tipo de preguntas, en un salón de clases con numerosos estudiantes, el proceso realizado por el docente para calificar las pruebas se vuelve dispendioso y muy lento, ocupando mucho espacio de su tiempo, encerrando al docente en tareas repetitivas e impidiendo que pueda dedicarle ese tiempo a otras actividades inherentes a su labor como docente, como la preparación de clase, investigación y/o la adecuada administración de las calificaciones.

Actualmente, la Universidad del Valle no cuenta con una alternativa de software basado en digitalización de exámenes, validación de la información y emisión de calificación. Este proyecto se enfoca en atender a los profesores que realicen evaluaciones en test o papel, que no cuenten con la conectividad a Internet estable y que les resulte complejo.

Teniendo en cuenta lo anterior, se plantea una alternativa que incluye un sistema integral de calificaciones que permita automatizar estas tareas, optimizando el tiempo y los demás recursos empleados por el docente para esta labor.

3. JUSTIFICACIÓN DE LA SOLUCIÓN PROPUESTA

Para solucionar el problema, se plantea que la herramienta de software a implementar apoye en gran medida la automatización del proceso realizado por el docente, brindando las siguientes ventajas:

- Reducir los costos de la captura manual de datos, liberando recursos para otras actividades docentes
- Mejora los tiempos de procesamiento de la información, contribuyendo a la productividad de la labor docente.
- Acelerar la entrega de resultados, al igual que permite el almacenamiento eficiente y el acceso a los datos para una adecuada administración de las calificaciones por parte del docente.
- Enviar por correo electrónico al profesor, el consolidado de los resultados de las evaluaciones, para así poder crear reportes y analizar datos sobre el rendimiento de los alumnos.
- Publicar los resultados en un documento en la red de Google Drive, para que pueda ser visualizado por los estudiantes a través de la nube.

Con esta herramienta se busca tener más precisión, rapidez y versatilidad, como respuesta a las necesidades de captura masiva de datos y procesamiento de información, así como también, se busca un ahorro significativo en el tiempo empleado por el docente para una labor tan importante como lo es, la revisión, evaluación y administración de calificaciones, si lo comparamos con la aplicación manual de dicha labor empleada actualmente por muchos docentes.

Esta solución planteada esta diseñada para un ambiente de escritorio y no un entorno web debido a la alimentación física de papel (exámenes) que se requiere al lado del cliente y no del servidor.

4. OBJETIVOS

4.1 Objetivo General

Desarrollar una herramienta de escritorio que permita ejercer una óptima gestión del proceso de calificación de exámenes, empleando la tecnología OMR a través del uso de un escáner y papel diseñado especialmente para este propósito.

4.2 Objetivos Específicos

- Realizar una interfaz gráfica para la configuración y entrada de datos, y la presentación de resultados obtenidos para su interpretación.
- Desarrollar un módulo de configuración de preguntas y respuestas del examen empleado por el profesor a evaluar
- Desarrollar un módulo de escaneo y revisión de cada uno de los exámenes realizados.
- Desarrollar un módulo de calificaciones en el cual se generen los resultados de los exámenes en una hoja de cálculo y se envíen al correo electrónico del profesor encargado y a Google Drive.

5. ALCANCE DEL PROYECTO

En este proyecto se contempla lo siguiente:

- La calificación automática de evaluaciones empleando la tecnología OMR
- La lectura óptica de la evaluación se realizará mediante un escáner con la biblioteca de manejo de periféricos SANE.
- No se guardará información del estudiante, del profesor, de las evaluaciones y sus respectivas calificaciones en ninguna base de datos a través de la aplicación del proyecto.
- Para realizar pruebas de campo, estas se harán en algunas asignaturas del profesor Ángel García Baños.

6. MARCO REFERENCIAL

6.1 Estado del arte.

En la actualidad, existen varias herramientas que me permiten evaluar exámenes de una forma automática. A continuación se presentan algunas aplicaciones:

- **CEA – TELEFORM**

Centro de Evaluación Automatizada (CEA), es una herramienta diseñada por InfoViews (empresa situada en la ciudad de México), para automatizar la calificación de evaluaciones utilizando un escáner y un software especializado en la captura de datos basado en el reconocimiento óptico de caracteres (ROC) , reconocimiento inteligente de caracteres (RIC) y el reconocimiento de marcas óptico (RMO). Estos métodos que disminuyen la cantidad de entradas de datos manual en ambientes de procesamiento de formularios y que le permiten a la aplicación aumentar su fiabilidad en la información recolectada.

CEA permite calificar de manera simultánea diferentes tipos de evaluaciones a diferentes grupos de estudiantes y de diferentes asignaturas, arrojando los resultados por evaluación en muy poco tiempo y exportándolos a la base de datos escolar. También contiene un módulo que permite diseñar e imprimir los exámenes en un formato especial para su captura de datos a través de un escáner. [1]

- **Remark Classic OMR:**

Creado por la empresa norteamericana SCANTRON, utiliza reconocimiento de marcas óptico (RMO) como base para la captura de los datos en exámenes, cuestionarios, encuestas y demás formularios que sean posibles de escanear, facilitando una evaluación automática de los mismos. Adicionalmente esta herramienta también permite capturar los datos de una manera instantánea y mostrar los resultados de los formularios en la pantalla de nuestra PC incluyendo tablas, gráficos y/o reportes de calificaciones, los cuales pueden ser posteriormente utilizados en bases de datos y paquetes estadísticos. Posee una interfaz muy intuitiva y amigable que permite comodidad y seguridad al usuario; sin embargo, no cuenta con un módulo para el diseño e impresión de exámenes. [2]

- **Evaldara DEF:**

De propiedad de la empresa española “DARA”, este software está diseñado para la gestión automatizada de exámenes tanto pruebas objetivas como psicométricas, generalizando este proceso. Las características de este software

permiten que la evaluación, validación y clasificación de los resultados se realicen en un único proceso. Trabaja con la metodología basada en lectura óptica empleando el RMO. Posee un módulo completamente adaptable que permite diseñar e imprimir las evaluaciones que se quieran realizar manejando un cifrado creado por la misma empresa que garantiza la identificación del examen impreso. Soporta muy bien el gran volumen de datos sin afectar su velocidad y precisión al generar los resultados. Esta aplicación se ha extendido en muchas universidades de España y algunas latinoamericanas como la escuela politécnica del ejército de Ecuador. [3]

6.2. Marco Conceptual

- RUBY

Es el lenguaje de programación base empleado en este proyecto. Es un lenguaje de programación interpretado, reflexivo y orientado a objetos. Fue creado por Yukihiro "Matz" Matsumoto, quien empezó a trabajar en Ruby el 24 de febrero de 1993, y lo presentó al público en el año 1995.

Ruby además de ser muy poderoso es sencillo, su sintaxis es simple y consistente, combina muchas características de otros lenguajes como Python, Perl y Smalltalk. Además, su implementación oficial es distribuida bajo una licencia de software libre [4]

- SANE

Scanner Access Now Easy (SANE), es una interfaz de programación de aplicaciones (API) que proporciona un acceso estándar a cualquier dispositivo de escaneo (escáner de sobremesa, escáner de mano, cámaras y videocámaras, etc.). El API de SANE es de dominio público y su discusión y desarrollo está abierto a todo el mundo. El código fuente actual está escrito para UNIX (incluye GNU/LINUX). [5]

- ICR ó RCI

“Intelligent Character Recognition” o en español “Reconocimiento de Caracteres Inteligente”. Es una tecnología que proporciona a los sistemas de reproducción por escáner y sistemas de imágenes la habilidad de convertir manuscritos(en letra no cursiva) en caracteres capaces de ser interpretados o reconocidos por una computadora. Así, las imágenes de caracteres en letra manuscrita son extraídas de un mapa de bits de la imagen reproducida por el escáner. [6]

- OCR ó ROC

“**Optical Character Recognition**” o en español “**Reconocimiento Óptico de Caracteres**”. Proporciona a los sistemas de reproducción por escáner y sistemas de imágenes la habilidad de convertir imágenes de caracteres en letra de máquina, a caracteres capaces de ser interpretados o reconocidos por una computadora. Así, las imágenes de caracteres en letra de máquina son extraídas de un mapa de bits de la imagen reproducida por el escáner. [7]

- OMR ó RMO

“**Optical Mark Recognition**” o en español “**Reconocimiento de Marcas Ópticas**”. Esta tecnología permite detectar la ausencia o presencia de una marca (hechas por humanos) desde documentos, como encuestas o exámenes. Se distingue del OCR en que no se necesita un reconocimiento de patrones complejo, es decir, existe poca probabilidad de leer incorrectamente. Por otra parte, requiere que la imagen tenga alto contraste y una forma fácil de reconocer. Una de las aplicaciones más comunes del OMR es el de exámenes múltiple opción en las universidades. [8]

- Google Drive

Google Drive, es un poderoso y fácil servicio de almacenamiento en la nube, el cual no tiene ningún costo o valor agregado, siempre y cuando no se exceda su capacidad inicial. Nos permite crear, compartir y guardar archivos en internet, los cuales además se pueden trabajar en tiempo real empleando Google Docs[9].

- Google Docs

Google Docs es una sencilla pero potente suite ofimática en la nube totalmente gratuita. Nos permite crear nuevos documentos, editar los que ya teníamos o compartirlos en la red. [10]

Dos de sus características más destacadas son:

- Los documentos se almacenan en línea: esto permite el acceso a ellos desde cualquier ordenador con conexión a internet, y compartirlos con cualquier persona que se desee, permitiendo su edición o llegar a publicarlos para que todo el mundo pueda verlos.
- La gran cantidad de formatos que soporta: con el procesador de texto podremos editar los documentos de texto y guardarlos con el mismo formato u otros distintos. Y lo mismo sucede con presentaciones y hojas de cálculo.

- Programación extrema (XP)

Programación extrema o su término en inglés “**Extreme Programming (XP)**”, es una de las metodologías de desarrollo de software más usadas en la actualidad para proyectos de corto plazo, equipo de trabajo de pocos integrantes y cuyo plazo de entrega es bastante corto. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. [11]

La metodología Extreme Programming se basa en:

- Pruebas Unitarias: Se basa en las pruebas realizadas a los principales procesos, de tal manera que adelantándonos en algo hacia el futuro, podamos hacer pruebas de las fallas que pudieran ocurrir. Es como si nos adelantáramos a obtener los posibles errores.
- Refabricación: Se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- Programación en pares: Consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento. Es como el chofer y el copiloto: mientras uno conduce, el otro consulta el mapa.
- Empieza en pequeño y añade funcionalidad con retroalimentación continua.
- El manejo del cambio se convierte en parte sustantiva del proceso.
- El costo del cambio no depende de la fase o etapa.
- No introduce funcionalidades antes que sean necesarias.
- El cliente o el usuario se convierte en miembro del equipo.

Para el óptimo desarrollo de la metodología, el cliente tiene unos deberes y/o opciones que son:

- Decidir que se implementa.
- Saber el estado real y el progreso del proyecto.
- Añadir, cambiar o quitar requerimientos en cualquier momento.
- Obtener lo máximo de cada semana de trabajo.
- Obtener un sistema funcionando cada 3 o 4 meses.

Por otro lado, el desarrollador tiene unos deberes y/o opciones que son:

- Decidir cómo se implementan los procesos.
- Crear el sistema con la mejor calidad posible.
- Pedir al cliente en cualquier momento aclaraciones de los requerimientos.
- Estimar el esfuerzo para implementar el sistema.
- Cambiar los requerimientos en base a nuevos descubrimientos.

En general, la metodología al ser de un desarrollo muy rápido se necesitan características fundamentales como la comunicación y retroalimentación,

concretas y frecuentes entre usuario final, desarrollador y cliente para poder realizar codificación y corrección casi al mismo tiempo.

- QT

QT es un conjunto de bibliotecas multiplataforma que son usadas ampliamente para el desarrollo de aplicaciones con interfaces gráficas de usuarios y sin interfaz gráfica como herramientas para servidores y consolas. Estas librerías utilizan el lenguaje de programación C++ de forma nativa, permitiendo que puedan ser utilizadas por muchos lenguajes como Ruby, Python, entre otros. [12]

7. METODOLOGÍA

En esta sección del documento se describen los artefactos más relevantes utilizados en el proceso de desarrollo de la herramienta de software, siguiendo la metodología de desarrollo programación extrema.

Esta metodología consta de 6 fases: exploración, planeación de la entrega, iteraciones, producción, mantenimiento y muerte. Para el desarrollo de esta aplicación se llevan a cabo las fases de exploración, planeación, diseño, codificación y pruebas, como se muestra en la ***Tabla 1***.

Durante la fase de exploración se estudiaron los requerimientos, metodologías y tecnologías de software necesarias para desarrollar por medio del uso de conceptos teóricos (representación y uso de OMR), una herramienta de software integrando una serie de conceptos empleados en aplicaciones con metodología OMR

En la fase de codificación se implementaron los requerimientos propuestos en este documento, y para la última fase se llevaron a cabo pruebas de caja negra del sistema con el objeto de comprobar el funcionamiento correcto de las funcionalidades desarrolladas.

Tabla 1. Fase de la metodología

Fase	Actividad
Exploración	•Orientación sobre las evaluaciones tipo test y metodología OMR.
Análisis o planeación	•Historias de usuario. •Tecnologías del proyecto.
Diseño	•Diseño computacional •Arquitectura del sistema
Producción	•Desarrollo en iteraciones 1. Primera iteración 2. Segunda iteración 3. Tercera iteración
Pruebas	•Pruebas del sistema

7.1 Fase de exploración

En esta fase, en términos generales, pretende proponer una forma eficiente de automatizar el proceso de calificación de exámenes tipo test, separándolo modularmente para que el examen pueda ser configurado, calificado y enviado a la nube.

Fue necesario estudiar los diferentes modos de exámenes tipo test así como la tecnología OMR empleada en diferentes evaluaciones.


7.2 Fase de análisis o planeación

7.2.1 Historias de usuarios

Las historias de usuarios son especificaciones poco formales de la visión del cliente con respecto a sus necesidades dentro del sistema, como se muestra en la **Tabla 2**:

- Actores: Usuarios que interactúan en el requerimiento planteado por el cliente.
- Iteración: Numeración de secuencia en la que se planea desarrollar la iteración.
- Estimación: Tiempo estimado para ver los resultados

Tabla 2. Historia de usuario No. 001

	HISTORIA DE USUARIO SISTEMA INTEGRAL DE CALIFICACIÓN AUTOMATIZADA	Iteración: 1
		No. 001
Titulo	Configuración del examen	
Actores	Profesor	
Descripción	Como profesor quiero configurar el examen indicando donde se encuentran las marcas en las esquinas de las hojas, el área donde se encontrará el código del estudiante e indicar donde se encuentran las opciones de las posibles respuestas a cada pregunta dentro del examen que se va a realizar a los estudiantes.	
Estimación	4 semanas	

Se relaciona en “Anexos I” todas las historias de usuarios que fueron necesarias para desarrollar este trabajo. Estas son: calificación del examen, visualización de resultados, exportar resultados y configuración general.

7.2.2 Tecnologías del proyecto

La metodología propuesta contempla una serie de estrategias a seguir y define las características de la tecnología a utilizar. Se estudiaron diferentes alternativas y se usaron las siguientes tecnologías:

- Lenguaje de programación Ruby.
- Diseño de interfaces con QT.
- Librerías SANE en la comunicación con periféricos.

7.3 Fase de diseño

7.3.1 Diseño computacional

En el diseño computacional se definió la estructura del software. Esta proporcionó un conjunto de modelos y diagramas que permitieron representar en profundidad los aspectos relacionados con los requisitos funcionales, no funcionales y riesgos planteados con los componentes del sistema, lo cual permitió modelar el sistema de manera que se satisficieran las necesidades y se mitigaran los riesgos. Además, permitió la división de los trabajos de implementación en partes más manejables y más fáciles de desarrollar.

7.3.1.1 Modelo de navegación

Este modelo proporciona la representación y documentación de la estructura navegable del sistema de forma sencilla y fácil de comprender, mostrando la forma como navegará cada usuario a través de la aplicación.

Se ilustrará un modelo de navegación describiendo el proceso e ilustrado en la **Figura 1:**

1. Ingreso a la aplicación de calificación de exámenes automatizados.
2. Ingreso al modulo de importar imagen o imágenes dependiendo de las hojas del examen.
3. Al importar la imagen, este mismo modulo entregará una respuesta verdadera (true) para formatos de imágenes válidas o una respuesta falsa(false) para formatos no soportados por el sistema o cancelar la elección de la imagen.
4. De acuerdo a la validación “true”: se continuará al formulario designado para la configuración de parámetros del examen, o a la validación “false”, retornará al usuario al panel de importar de imagen.
5. Configurar y almacenar los parámetros del examen en un archivo.

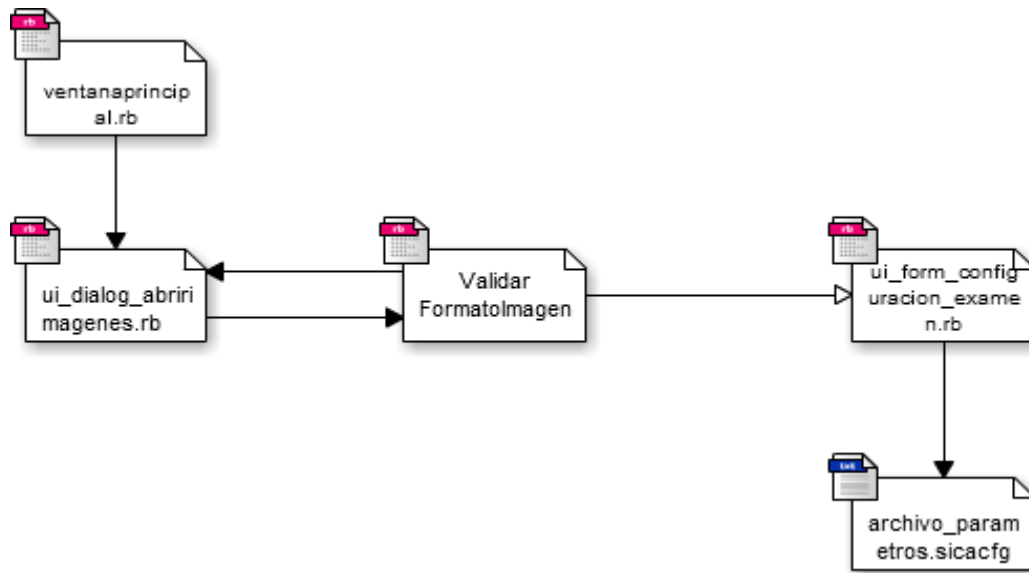


Figura 1. Modulo configuración examen – Modelo de navegación

Se relacionaran los modelos de navegación en “Anexos II” para: Guardar configuración de examen, evaluar examen, configuración de parámetros generales, subir archivos a Google Drive.

7.3.2 Arquitectura del sistema

El proyecto cuenta con una arquitectura general dividida en una parte física y lógica. La física presenta la estructura necesaria para el funcionamiento del sistema y la lógica presenta las principales características del diseño del sistema a nivel de software. En la arquitectura del sistema se definen las vistas de la aplicación basadas en los patrones y los requerimientos funcionales y no funcionales del proyecto.

Además, sirve como guía para definir las capas de la aplicación junto con el software y hardware necesario, decisión fundamental, ya que debe seleccionarse de acuerdo a las expectativas de funcionamiento y agilidad que se desee.

Con el fin de separar la lógica del diseño dentro de la aplicación de escritorio, se buscó desarrollar una aplicación estructurada y ágil, implementando una vista física y vista lógica.

A continuación se muestra detalladamente las vistas arquitecturales del software:

7.3.2.1 Vista física

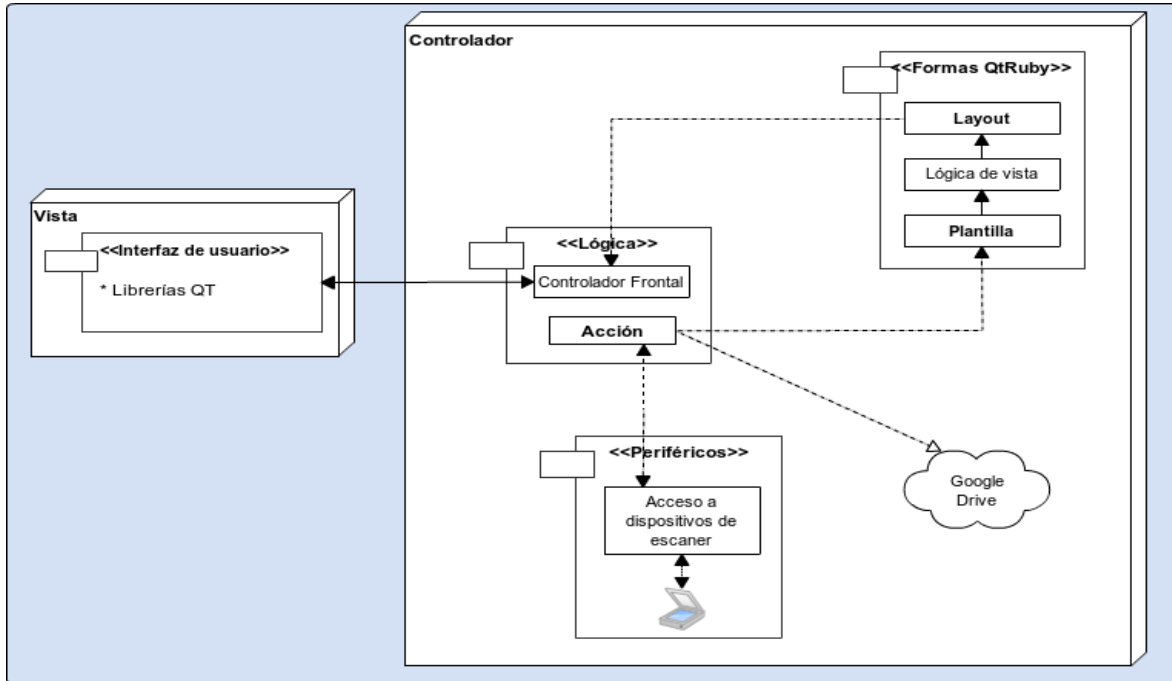


Figura 2. Vista física de la aplicación

Para lograr un mejor desempeño en la aplicación, se plantea una arquitectura que incluye el uso de:

- Un equipo de cómputo, sea de escritorio o portátil.
- Un dispositivo de escaneo de imágenes.

Las características en hardware para el equipo de cómputo y el escáner son las siguientes:

Equipo de cómputo:

- Mínimo 512 mega-bytes de memoria RAM.
- Monitor, teclado, ratón.
- Procesador (AMD o Intel) 2.0 GHZ o más.
- Conexión a internet (solo si se quiere enviar algún archivo a la nube)

Dispositivo escáner:

- De alta velocidad.
- Alimentador automático.

7.3.2.2 Vista lógica

La vista lógica presenta las principales características del diseño del sistema a nivel de software, está representada por capas de software que intervienen en la implementación de la funcionalidad del sistema.

Las características en software para para el equipo de computo y el escáner son las siguientes:

Equipo de computo:

- Contar con un sistema operativo Linux de 32 bits.
- Librerías QT y complemento “qtuitools”.
- Ruby versión 1.8.7 .
- Adicionar las gemas de Ruby: “sane”, “ffi”, “rubygems”, “qtruby4”, “nokogiri (versión 1.5.5 o superior)”, “roo (versión 1.10.1 o superior) ”, “ruby-gmail”, “mime”.

Dispositivo escáner:

- Compatible con Linux.
- Compatible con SANE.

A continuación se describen las capas de software de esta vista lógica:

- **Capas de software**

El sistema se organiza en dos capas de software: La primera es la capa de interfaz de usuario (Vista) y la segunda es la capa de lógica (Controlador).

Capa de interfaz de usuario:

Esta capa procesará las órdenes realizadas a través de la interfaz y las enviará al controlador. Es implementada con ayuda de las librerías de QT. De esta manera se permite manejar aspectos de presentación.

Capa lógica:

Responsable de responder las órdenes enviadas por el usuario a través de la interfaz. Este componente encapsula la lógica necesaria para realizar los procesos que conlleva configurar un examen, además del manejo de periféricos y comunicación con la nube.

Esta capa es implementada en el lenguaje de programación Ruby.

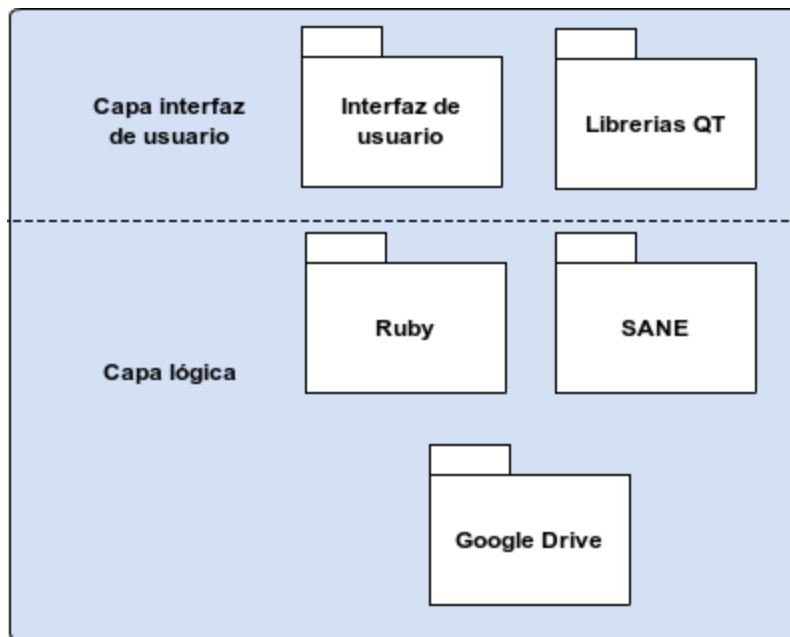


Figura 3. Vista lógica de la aplicación

7.4 Fase de producción

Una vez cumplidas las fases de exploración, análisis y diseño en las cuales se estableció la estructura del software, se procede a implementar las funcionalidades propuestas.

Para cada iteración se establece la funcionalidad esperada y se describe a grandes rasgos la forma en que se llevó a cabo el desarrollo. Para obtener validación de resultados en estas, se realizaron pruebas unitarias a cada uno de los formularios QT asociados, para comprobar que cada uno de los mismos es confiable.

A continuación se describe el desarrollo de la herramienta de software en 3 iteraciones.

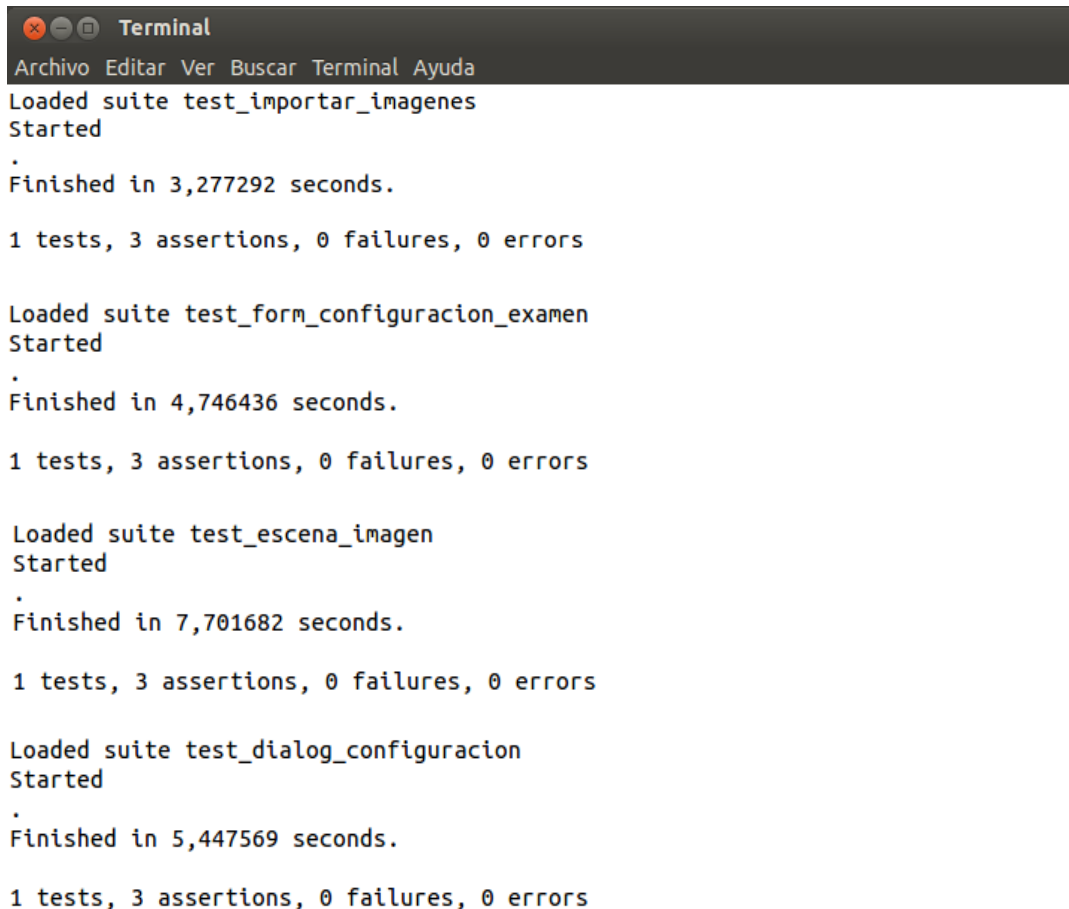
7.4.1 Primera iteración

Funcionalidad esperada:

- Configuración del examen para el estudiante.

Para llevar a cabo se desarrollaron formularios de ingreso de información e imágenes, implementados con las librerías QT, la cual permite mediante controladores diseñados en Ruby, validar los parámetros a guardar de la configuración, los cuales son almacenados en un archivo de texto.

Las pruebas unitarias se realizaron con el framework Test::Unit, de Ruby. Los resultados se visualizan a continuación en la ***Figura 4***:



```
Terminal
Archivo Editar Ver Buscar Terminal Ayuda
Loaded suite test_importar_imagenes
Started
.
Finished in 3,277292 seconds.

1 tests, 3 assertions, 0 failures, 0 errors

Loaded suite test_form_configuracion_examen
Started
.
Finished in 4,746436 seconds.

1 tests, 3 assertions, 0 failures, 0 errors

Loaded suite test_escena_imagen
Started
.
Finished in 7,701682 seconds.

1 tests, 3 assertions, 0 failures, 0 errors

Loaded suite test_dialog_configuracion
Started
.
Finished in 5,447569 seconds.

1 tests, 3 assertions, 0 failures, 0 errors
```

Figura 4. Pruebas unitarias primera iteración.

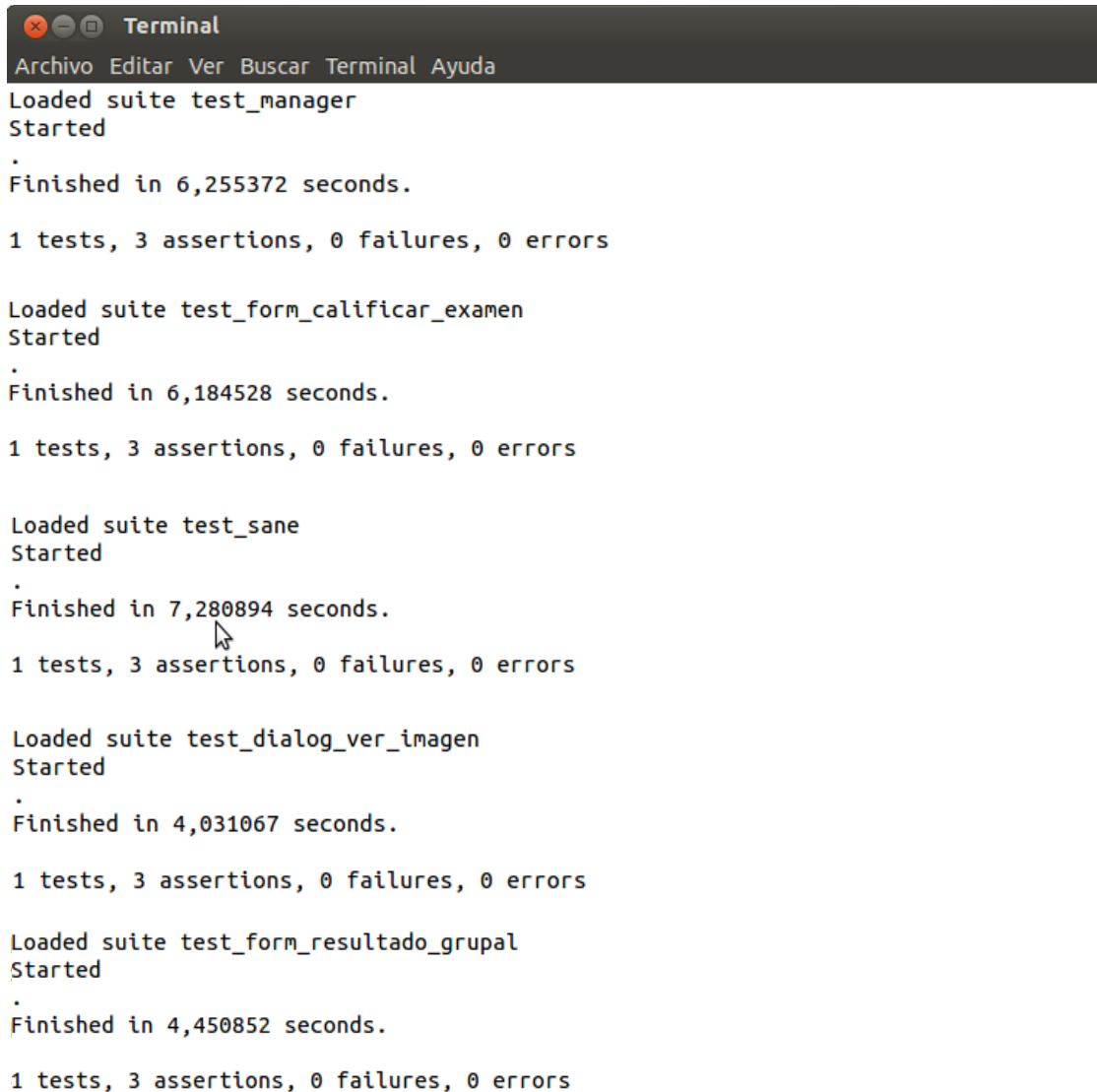
7.4.2 Segunda iteración

Funcionalidad esperada:

- Calificación del examen realizado al estudiante.

Para llevar a cabo se implementaron controladores en Ruby y algunas interfaces con librerías QT para la manipulación de imágenes, dispositivos de escaneo, visualización de resultados sin necesidad de exportar y visualización de los parámetros configurados en la iteración anterior.

Al igual que la iteración anterior, las pruebas se realizaron con el framework Test::Unit, de Ruby. Los resultados se visualizan a continuación en la **Figura 5**:



```
Terminal
Archivo Editar Ver Buscar Terminal Ayuda
Loaded suite test_manager
Started
.
Finished in 6,255372 seconds.

1 tests, 3 assertions, 0 failures, 0 errors

Loaded suite test_form_calificar_examen
Started
.
Finished in 6,184528 seconds.

1 tests, 3 assertions, 0 failures, 0 errors

Loaded suite test_sane
Started
.
Finished in 7,280894 seconds.

1 tests, 3 assertions, 0 failures, 0 errors

Loaded suite test_dialog_ver_imagen
Started
.
Finished in 4,031067 seconds.

1 tests, 3 assertions, 0 failures, 0 errors

Loaded suite test_form_resultado_grupal
Started
.
Finished in 4,450852 seconds.

1 tests, 3 assertions, 0 failures, 0 errors
```

Figura 5. Pruebas unitarias segunda iteración.

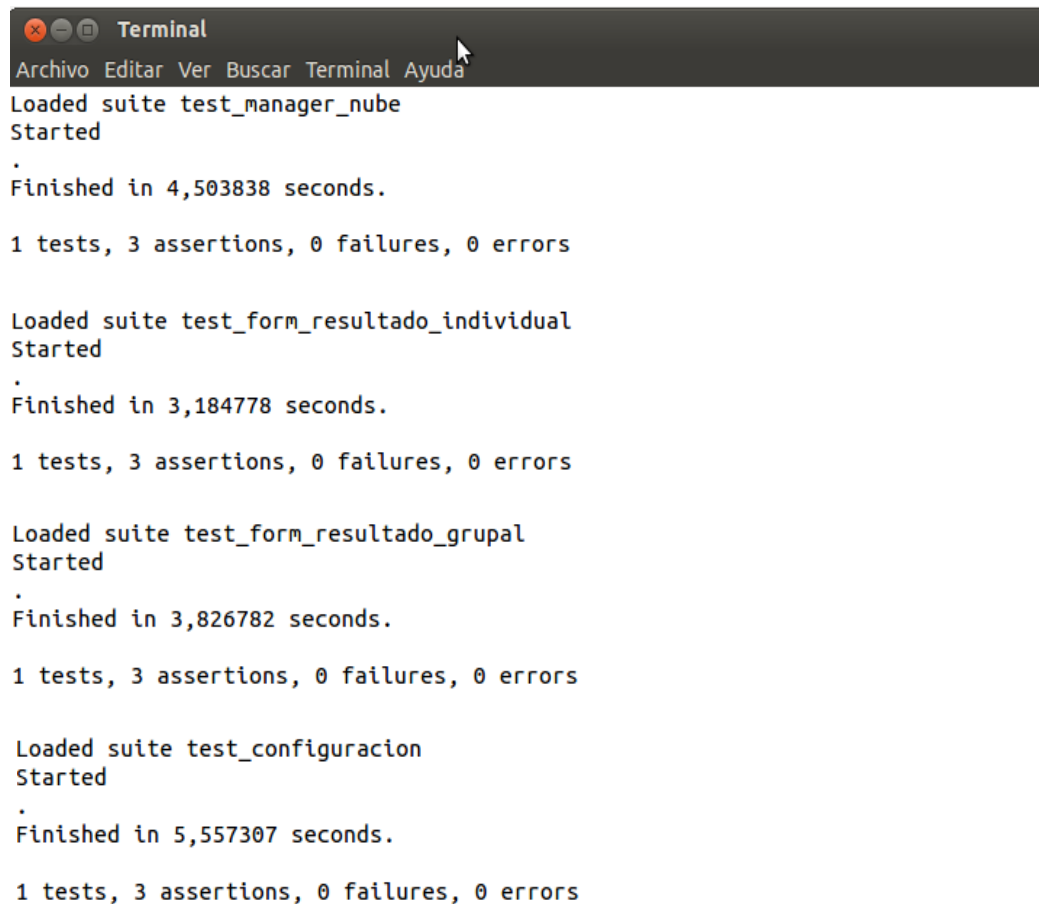
7.4.3 Tercera iteración

Funcionalidad esperada:

- Exportar resultados de las calificaciones realizadas al examen.

Para llevar a cabo se implementaron controladores en Ruby para la manipulación de tablas en hojas de Excel y comunicación con la nube y Google Docs, a partir de los resultados arrojados en la iteración anterior.

Al igual que en las anteriores iteraciones, las pruebas se realizaron con el framework Test::Unit, de Ruby. Los resultados se visualizan a continuación en la **Figura 6**:



```
Terminal
Archivo Editar Ver Buscar Terminal Ayuda
Loaded suite test_manager_nube
Started
.
Finished in 4,503838 seconds.

1 tests, 3 assertions, 0 failures, 0 errors

Loaded suite test_form_resultado_individual
Started
.
Finished in 3,184778 seconds.

1 tests, 3 assertions, 0 failures, 0 errors

Loaded suite test_form_resultado_grupal
Started
.
Finished in 3,826782 seconds.

1 tests, 3 assertions, 0 failures, 0 errors

Loaded suite test_configuracion
Started
.
Finished in 5,557307 seconds.

1 tests, 3 assertions, 0 failures, 0 errors
```

Figura 6. Pruebas unitarias tercera iteración.

Se anexan los archivos utilizados para realizar las pruebas unitarias mencionadas en cada una de las iteraciones anteriores.

7.5 Fase de pruebas

En la **Tabla 3** se muestra el formato para las pruebas del sistema. En la columna izquierda están los ítem a tener en cuenta, y en la columna derecha se colocarán los resultados de éstas.



	PRUEBAS DEL SISTEMA	PS: 1
	SISTEMA INTEGRAL DE CALIFICACIÓN AUTOMATIZADA	No. ###
Fecha de Realización:	(DD-MM-AAAA) -- Fecha en que se realizó la prueba.	
Módulo de Referencia:	Es el módulo al que pertenece la prueba que se está realizando.	
Requerimientos:	Requerimientos a probar.	
Objetivos de la prueba:	Se explica de forma precisa cuál va a ser la funcionalidad de la prueba, es decir se describe lo que se quiere probar.	
Procedimiento:	Se describe en forma secuencial, los pasos que se deben seguir para cumplir los objetivos de la prueba.	
Datos de Prueba:	Campo: Dato -- Aquí se detallan las entradas que van a ser probadas como por ejemplo los campos de un formulario junto con lo que el responsable de la prueba debe de ingresar para que sea procesada.	
Resultados esperados:	Aquí se describe lo que el sistema respondería si su funcionalidad es correcta.	
Resultados obtenidos:	Prueba Exitosa SI () NO () -- Responde si la prueba fue correcta. Observaciones: -- Aquí se colocan las respuesta obtenidas sin tener en cuenta que fue exitosa o no y las observaciones necesarias.	
Responsables de la Prueba:	Nombre y apellido -- Aquellos usuarios encargados de realizar las pruebas.	
Observación:	Comentario de la prueba realizada.	

Tabla 3. Modelo de prueba del sistema No. ###

7.5.1 Pruebas del sistema

	PRUEBAS DEL SISTEMA	PS: 1
	SISTEMA INTEGRAL DE CALIFICACIÓN AUTOMATIZADA	No. 001
Fecha de Realización:	13-Noviembre-2012	
Módulo de Referencia:	Modulo Importar Examen	
Requerimientos:	<ul style="list-style-type: none"> • Importar la imagen del examen en blanco a realizar a los estudiantes. • Importar la imagen o imágenes de los exámenes calificados de los estudiantes. 	
Objetivos de la prueba:	Esta prueba se realiza con el fin de demostrar que el sistema está en la capacidad importar la imagen de los exámenes calificados o examen en blanco para configurar, desde dispositivos de escaneo o archivos ubicados en el disco duro.	
Procedimiento:	<ul style="list-style-type: none"> • Ingresar al formulario de configuración de examen o calificación de examen. • Elegir el archivo o los archivos a importar si se va a escoger desde el disco duro. Si se va a escanear, indicar el número de hojas a tomar del alimentador de documentos para enviar la orden al dispositivo. • Clic en el botón “Aceptar”, si se van a importar imágenes locales y toma la ruta de las imágenes del campo de texto; o clic en “Escanear” si se obtuvieron las imágenes directamente del escáner, las cuales son almacenadas en una carpeta local con un nombre estándar “filescaan_anomesdiahorasminutossegundos”. • Clic en el botón “Aceptar” al obtener la o las imágenes a trabajar. 	
Datos de Prueba:	<p>Importar imagen desde disco duro:</p> <ul style="list-style-type: none"> • Campo de texto (path) : (Puedes seleccionar una o más imágenes) “/tmp/filescaan_20121110114414.pnm” <p>Importar imagen desde escáner:</p> <ul style="list-style-type: none"> • Escanear páginas deseadas. • Toma imagenes automáticamente de la carpeta local “filescaan_20121110114714” 	

Resultados esperados:	Obtención de la imagen a trabajar y pasarla al formulario de configuración o calificación del examen.
Resultados obtenidos:	Prueba Exitosa SI (X) NO ()
Responsables de la Prueba:	Fabián Payan
Observación:	

Tabla 4. Prueba del sistema No. 001

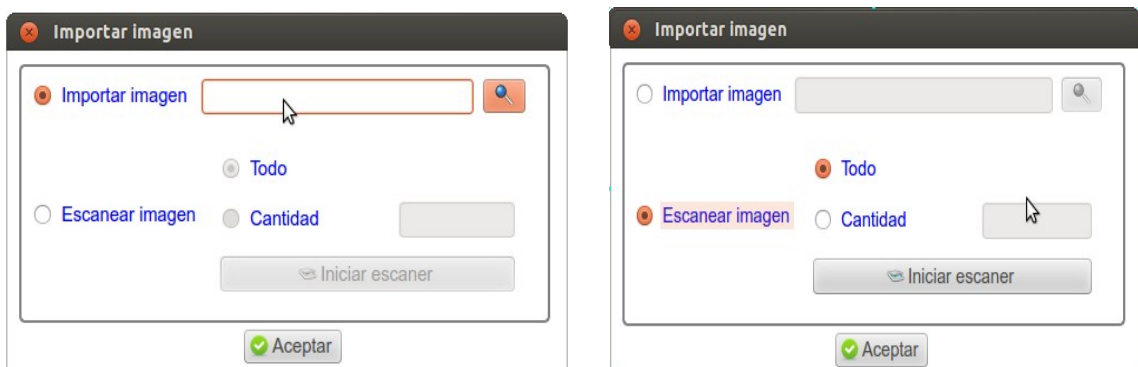


Figura 7. Captura de pantalla: Prueba modulo Importar Imagen desde disco duro (izquierda) y modulo Importar Imagen desde el escáner (derecha).

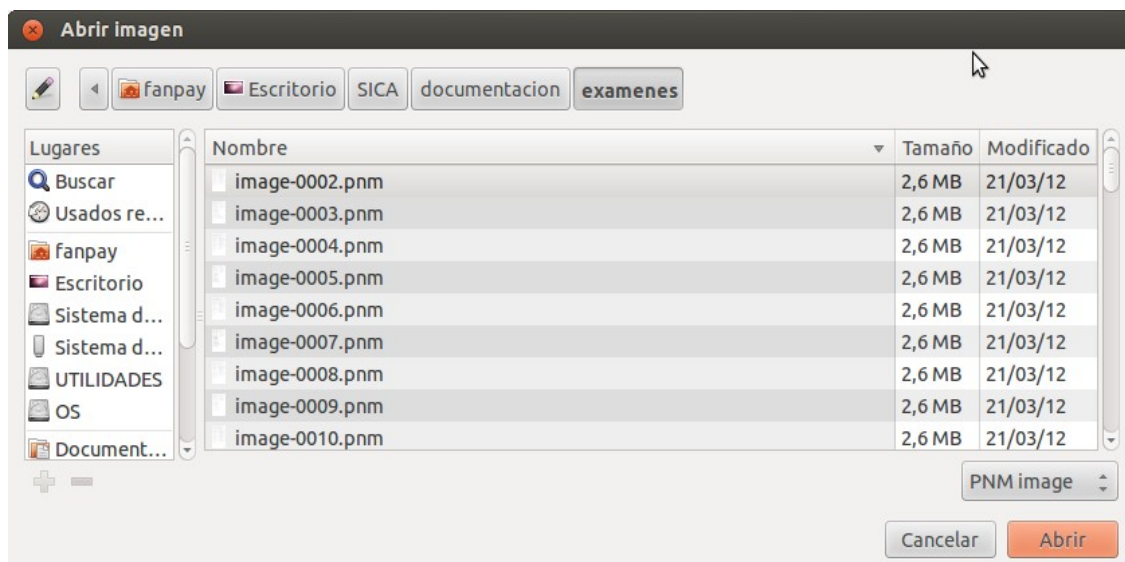


Figura 8. Captura de pantalla: Prueba modulo Importar Imagen desde disco duro. Buscando imagen o imágenes a importar.

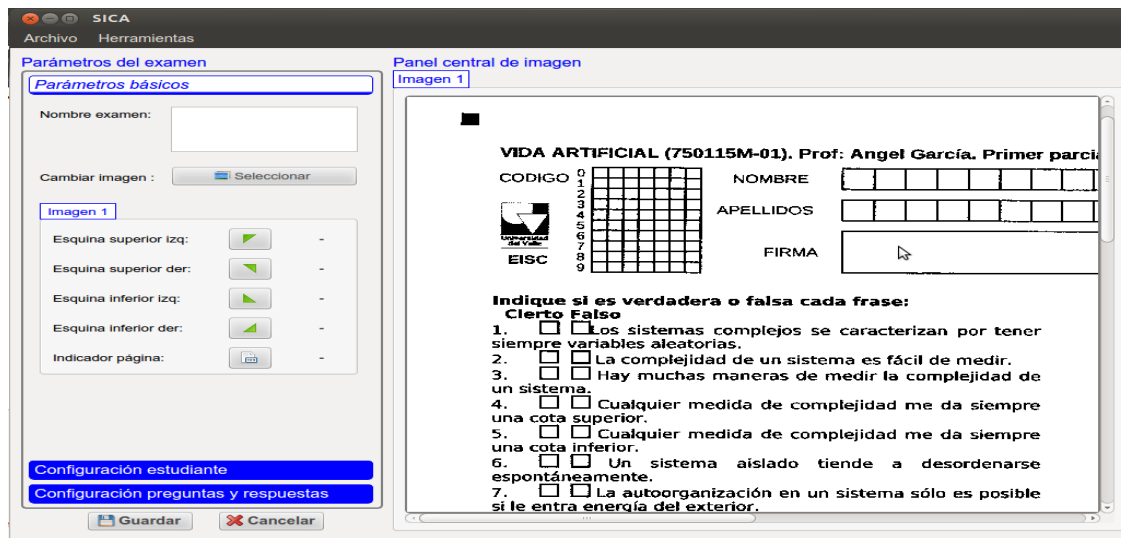


Figura 9. Captura de pantalla: Prueba modulo Configurar Examen con la imagen obtenida.

Se relacionan las pruebas del sistema en “Anexos III” para: Configuración del examen, calificación del examen realizado, exportar resultados, configuración de parámetros generales.

Para las pruebas realizadas directamente con el escáner se utilizó el formato de evaluación relacionado en “ANEXOS IV”.


8. CONCLUSIONES


- Con el proyecto se logro realizar una aplicación de escritorio que puede comunicarse con dispositivos de escaneo y exportar archivos a la nube, aprovechando la integración de una tecnología actual muy fuerte como Ruby.
- La integración de tecnologías como Ruby y librerías QT junto a la interfaz de programación SANE para la comunicación universal de periféricos, permitieron desarrollar la aplicación de tal manera que cumpliera con los objetivos planteados.
- Desarrollando la aplicación con Ruby, he conocido la filosofía y algunas características de este lenguaje como su versatilidad, poder, comodidad y fácil aprendizaje. Considero que este lenguaje posee un futuro grandioso y se expandirá rápidamente a medida que se explore más su potencial.
- Las interfaces fueron diseñadas utilizando la información más relevante para el profesor, permitiendo facilitar su comprensión y siendo amigable para él, además de agilizar sus procesos en cuanto a calificación de exámenes tipo test se refiere.
- Al aplicar la metodología XP durante el proceso de desarrollo, se obtuvo un mayor control en la generación de los errores y mayor claridad en la finalidad de los objetivos.


9. REFERENCIAS BIBLIOGRÁFICAS


- [1] CEA, Centro de evaluaciones automáticas, disponible en:
<http://www.infoviews.com.mx/CEA/> - [Consulta: 15 de Noviembre de 2011]
- [2] Remark Classic OMR, disponible en:
<http://espanol.scantron.com/software/remark.htm> - [Consulta: 15 de Noviembre de 2011]
- [3] Grupo DARA innovación y tecnología, disponible en: <http://www.dara.es/omr/> - [Consulta: 18 de Noviembre de 2011]
- [4] Ruby a programmer's best friends, disponible en: <http://www.ruby-lang.org/es> - [Consulta: 3 de Noviembre de 2011]
- [5] SANE, Scanner Access Now Easy, disponible en: <http://www.sane-project.org/> - [Consulta: 3 de Noviembre de 2011]
- [6] Wikipedia, la enciclopedia libre, ICR, disponible en:
http://en.wikipedia.org/wiki/Intelligent_character_recognition - [Consulta: 18 de Diciembre de 2011]
- [7] Wikipedia, la enciclopedia libre, OCR, disponible en:
http://es.wikipedia.org/wiki/Reconocimiento_optico_de_caracteres - [Consulta: 18 de Diciembre de 2011]
- [8] OptionScan, especialistas en OMR y Microfilm/Digitalización, disponible en:
<http://opcionscan.com/omr.htm> - [Consulta: 18 de Diciembre de 2011]
- [9] Google Drive, Visión general de Google Drive, disponible en:
<http://support.google.com/a/bin/answer.py?hl=es&hlrm=es&answer=2490026> - [Consulta: 12 De Agosto de 2012]
- [10] Google Docs, online documents with real-time collaboration, disponible en:
<http://www.google.com/apps/intl/es/business/docs.html> - [Consulta: 3 De Enero de 2012]
- [11] BECK, Kent. "Extreme Programming Explained. Embrace Change", Pearson Education, 1999. Traducido al español como: "Una explicación de la programación Extrema". Aceptar el cambio", Addison Wesley, 2000
- [12] Qt Developer Network, Get Started, disponible en: <http://qt-project.org/> - [Consulta: 14 de Abril de 2012]
- [13] THOMAS, Kent; FOWLER, Chad; HUNT, Andy . "Programming Ruby, Second Edition", The Pragmatic Programmers, 2005.

ANEXOS I

	HISTORIA DE USUARIO SISTEMA INTEGRAL DE CALIFICACIÓN AUTOMATIZADA	Iteración: 2
		No. 002
Titulo	Calificación del examen	
Actores	Profesor	
Descripción	Como profesor quiero una herramienta que pueda calificar automáticamente uno o más exámenes realizados a los estudiantes de acuerdo a la configuración correspondiente.	
Estimación	4 semanas	

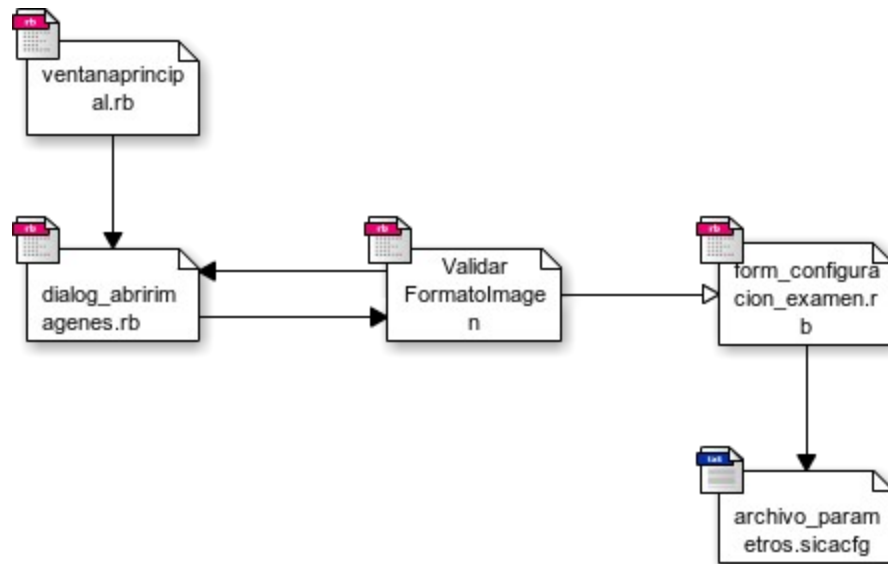
	HISTORIA DE USUARIO SISTEMA INTEGRAL DE CALIFICACIÓN AUTOMATIZADA	Iteración: 2
		No. 003
Titulo	Visualización de resultados	
Actores	Profesor	
Descripción	Como profesor quiero visualizar los resultados de los exámenes calificados ya sea uno o más exámenes, en una tabla en la aplicación.	
Estimación	1 semana	

	HISTORIA DE USUARIO SISTEMA INTEGRAL DE CALIFICACIÓN AUTOMATIZADA	Iteración: 2,3
		No. 004
Titulo	Exportar resultados	
Actores	Profesor	
Descripción	Como profesor quiero exportar los resultados de las evaluaciones realizadas a un formato en hoja de cálculo y/o subirlo a la nube de información.	
Estimación	1 semanas	

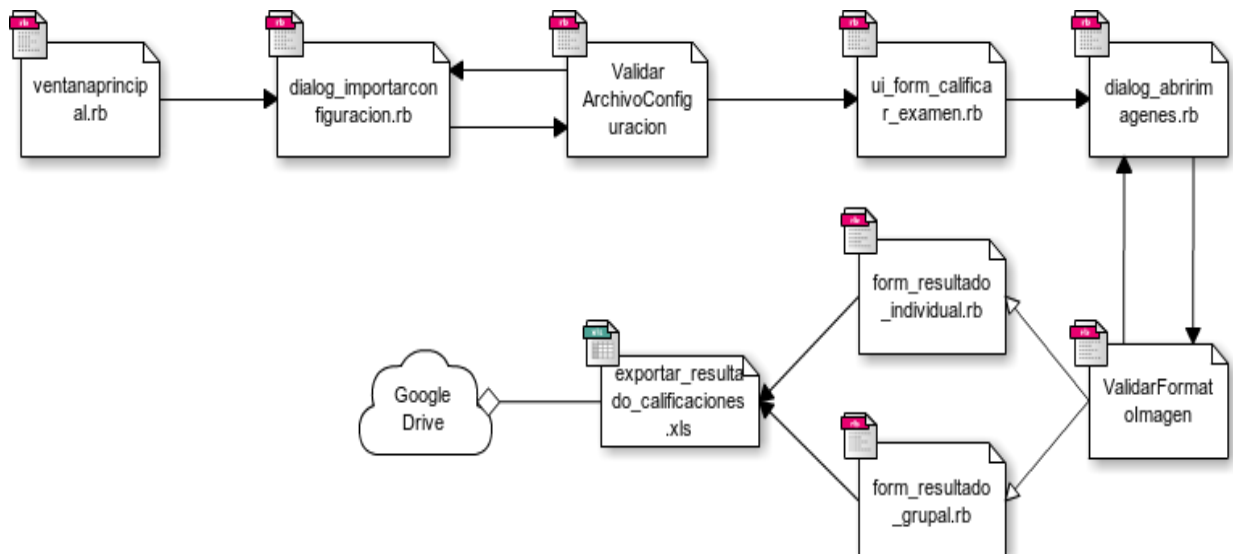
	HISTORIA DE USUARIO SISTEMA INTEGRAL DE CALIFICACIÓN AUTOMATIZADA	Iteración: 1,2,3
		No. 005
Titulo	Configuración general	
Actores	Profesor	
Descripción	Como profesor quiero ser capaz de configurar las dimensiones de las marcas de las respuestas y elegir el dispositivo por el cual se van a escanear las evaluaciones.	
Estimación	1 semanas	

ANEXOS II

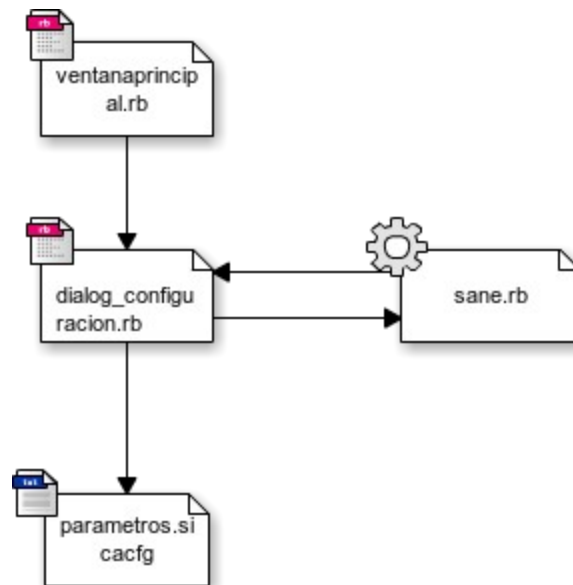
1. Guardar configuración de examen



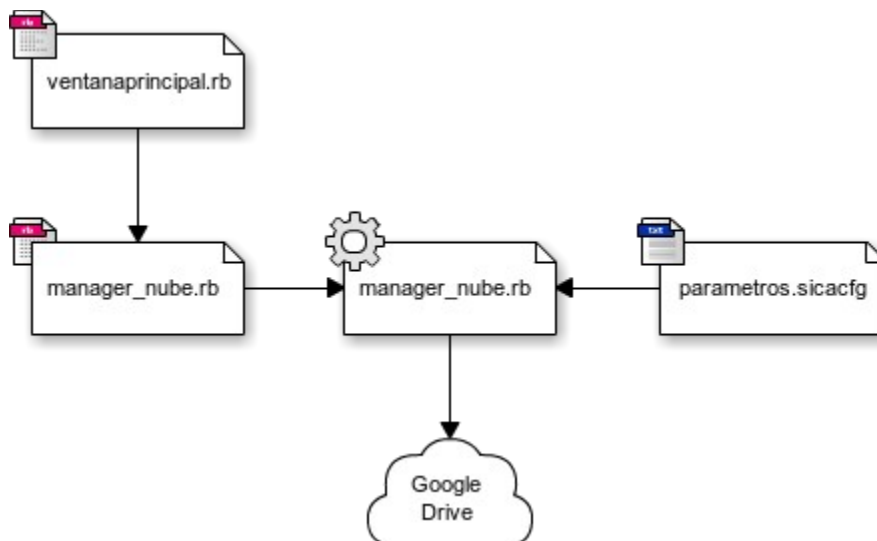
2. Evaluar examen




3. Configuración de parámetros generales



4. Subir archivos a "Google Drive"



ANEXOS III

	PRUEBAS DEL SISTEMA	PS: 2
	SISTEMA INTEGRAL DE CALIFICACIÓN AUTOMATIZADA	No. 002
Fecha de Realización:	13-Noviembre-2012	
Módulo de Referencia:	Modulo Configurar Examen	
Requerimientos:	<ul style="list-style-type: none"> • Importar la imagen del examen en blanco a realizar a los estudiantes. • Configurar marcas de orientación de la hoja. • Configurar parámetros de ubicación del código de estudiante. • Configuración de preguntas y respuestas. 	
Objetivos de la prueba:	Esta prueba se realiza para demostrar que la aplicación está en capacidad de configurar un examen, indicando sus marcas de orientación, área del código de estudiante e indicando las preguntas y respuestas que conlleva el examen. Estos parámetros son almacenados en un archivo de texto el cual posteriormente se utilizará para evaluarlo.	
Procedimiento:	<ul style="list-style-type: none"> • Obtener la imagen del módulo “Importar Examen”, que se abre automáticamente al ingresar en el formulario de configuración de examen. • La imagen se visualizara en un panel especial, para que puedan ser manipulada. • Configurar el nombre que va a llevar el examen, y seleccionar la ubicación de sus marcas de orientación y paginación. • Clic en la sección “Configurar estudiante”, para indicar el área donde se ubicará el código de los estudiantes. • Clic en la sección “Configurar preguntas/respuestas”, para indicar el área de cada pregunta y respuesta tipo test realizada en el examen que quiera ser evaluada. Cada respuesta se le puede adherir un valor. • Clic en el botón “Guardar” para almacenar los parámetros en un archivo con extensión (*.sicacfg). 	
Datos de Prueba:	<ul style="list-style-type: none"> • Obtener la imagen “/tmp/filescan_20121110114414.pnm” • Se determina el número de páginas, en este caso 1. • Marcar las esquinas superior izquierda, superior derecha, inferior izquierda e inferior derecha. 	

	<ul style="list-style-type: none"> • Selección de área del código de estudiante. • Preguntas configuradas: 1,2,3 • Tipo de preguntas: Única respuesta. • Número de respuestas por pregunta: 2 • Nombre del archivo a almacenar: "examen_vidaartificial.sicacfg".
Resultados esperados:	Recibir la imagen a trabajar, pasarla al formulario de configuración y almacenar exitosamente los parámetros configurados.
Resultados obtenidos:	Prueba Exitosa SI (X) NO ()
Responsables de la Prueba:	Fabián Payan
Observación:	

SICA

Examen Herramientas

Parámetros del examen

Panel central de imagen

Parámetros básicos

Nombre examen: Examen vida artificial No. 2

Cambiar imagen : Seleccionar

Imagen 1

Esquina superior izq: OK

Esquina superior der: -

Esquina inferior izq: -

Esquina inferior der: -

Indicador página: -

Configuración estudiante

Configuración preguntas y respuestas

Guardar

Cancelar

Imagen 1

VIDA ARTIFICIAL (750115M-01). Prof: Angel García. Primer parcial

CODIGO

0

1

2

3

4

5

6

7

8

9

U

N

I

V

E

R

S

I

D

A

D

A

D

EISC

NOMBRE

APELLIDOS

FIRMA

Indique si es verdadera o falsa cada frase:

Cierto

Falso

1.

Los sistemas complejos se caracterizan por tener siempre variables aleatorias.

2.

La complejidad de un sistema es fácil de medir.

3.

Hay muchas maneras de medir la complejidad de un sistema.

4.

Cualquier medida de complejidad me da siempre una cota superior.

5.

Cualquier medida de complejidad me da siempre una cota inferior.

6.

Un sistema aislado tiende a desordenarse espontáneamente.

37

SICA

Examen Herramientas

Parámetros del examen

Parámetros básicos

Configuración estudiante

Imagen 1

Seleccionar código estudiante

Capturar

Area

168,0 _ 140,0

seleccionada:

270,0 _ 288,0

Configuración preguntas y respuestas

Guardar

Cancelar

Panel central de imagen

Imagen 1

VIDA ARTIFICIAL (750115M-01). Prof: Angel García. Primer parcial

CODIGO

0

1

2

3

4

5

6

7

8

9

NOMBRE

APELLIDOS

FIRMA

Indique si es verdadera o falsa cada frase:

Cierto Falso

1.

Los sistemas complejos se caracterizan por tener siempre variables aleatorias.

2.

La complejidad de un sistema es fácil de medir.

3.

Hay muchas maneras de medir la complejidad de un sistema.

4.

Cualquier medida de complejidad me da siempre una cota superior.

5.

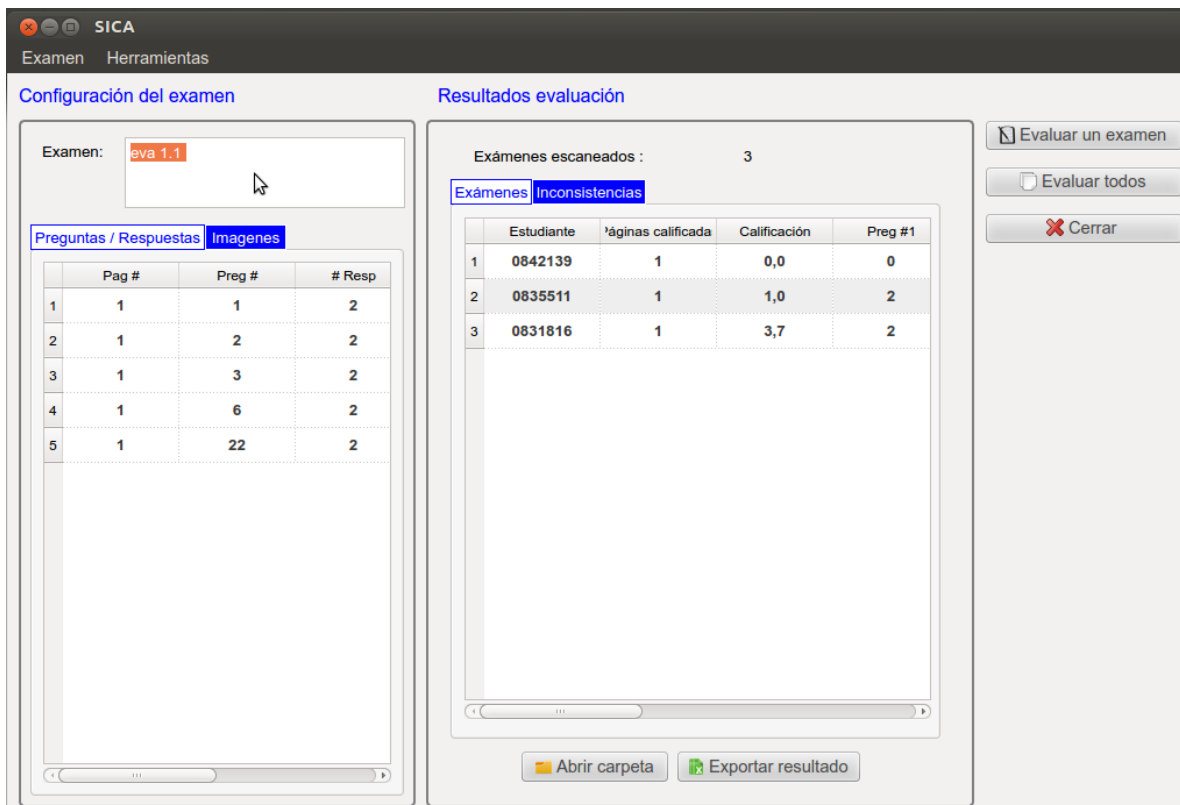
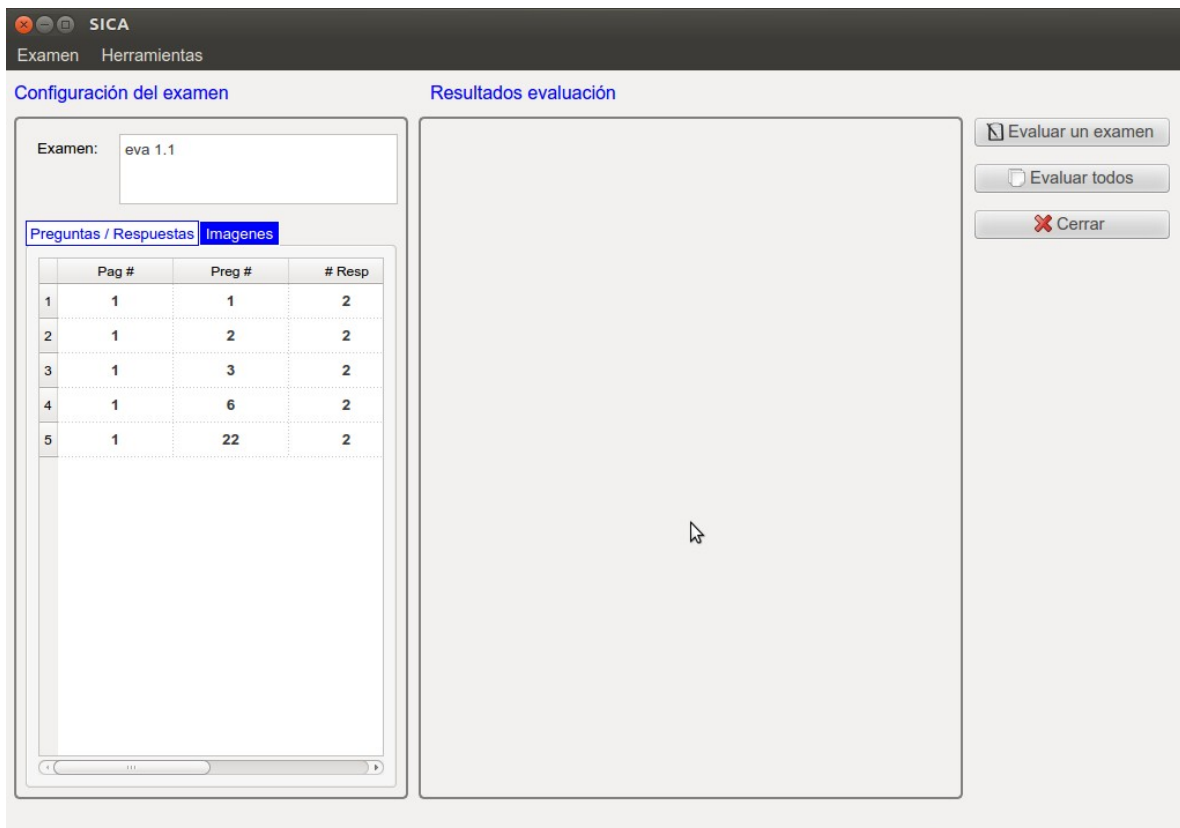
Cualquier medida de complejidad me da siempre una cota inferior.


6.

Un sistema aislado tiende a desordenarse


38


	PRUEBAS DEL SISTEMA	PS: 3
	SISTEMA INTEGRAL DE CALIFICACIÓN AUTOMATIZADA	No. 003
Fecha de Realización:	14-Noviembre-2012	
Módulo de Referencia:	Modulo Evaluar Examen	
Requerimientos:	<ul style="list-style-type: none"> • Importar el archivo de parámetros configurados para evaluar el examen y visualizarlos. • Importar examen o exámenes de uno o más estudiante a calificar. • Visualizar resultados, con sus calificaciones respectivas. 	
Objetivos de la prueba:	Esta prueba se realiza para demostrar que la aplicación está en capacidad de evaluar las imágenes escaneadas de uno o más exámenes realizados por los estudiantes. La evaluación se hace de acuerdo a los parámetros importados previamente configurados en el modulo “Configurar Examen”.	
Procedimiento:	<ul style="list-style-type: none"> • Importar archivo de parámetros del examen a evaluar. • Dar clic en “Evaluar examen” para importar imagen o imágenes de los exámenes de los estudiantes a evaluar, a través del modulo “Importar Examen” • Visualizar en una tabla, las preguntas con las respectivas respuestas del estudiante, su valoración por cada pregunta. 	
Datos de Prueba:	<ul style="list-style-type: none"> • Archivo de parámetros del examen :”vidaartificial.sicacfg” • Se importaron 3 imágenes de exámenes directamente desde el escáner. 	
Resultados esperados:	Visualizar los resultados de los exámenes calificados en una tabla, identificados por cada estudiante.	
Resultados obtenidos:	Prueba Exitosa SI (X) NO ()	
Responsables de la Prueba:	Fabián Payan	
Observación:		



	PRUEBAS DEL SISTEMA	PS: 4
	SISTEMA INTEGRAL DE CALIFICACIÓN AUTOMATIZADA	No. 004
Fecha de Realización:	14-Noviembre-2012	
Módulo de Referencia:	Modulo Exportar Resultados	
Requerimientos:	<ul style="list-style-type: none"> • Evaluar uno o más exámenes de estudiantes correctamente. • Exportar las calificaciones en un archivo de formato Excel, o enviarlas a Google Drive. 	
Objetivos de la prueba:	Esta prueba se realiza para demostrar que la aplicación está en capacidad de exportar los resultados de las calificaciones evaluadas a un formato de hoja de cálculo y/o enviar el archivo a la nube.	
Procedimiento:	<ul style="list-style-type: none"> • Evaluar uno o más exámenes a través del modulo “Evaluar Examen”. • Visualizar en una tabla, las preguntas con las respectivas respuestas del estudiante, su valoración por cada pregunta. • Dar clic en “Exportar resultado” y elegir si solo se va a crear el archivo Excel o se va a subir a la nube directamente. 	
Datos de Prueba:	<ul style="list-style-type: none"> • Archivo de parámetros del examen :”vidaartificial.sicacfg” • Se importaron 3 imágenes directamente desde el escáner. • Nombre del archivo a exportar: “Calificaciones VA 2012.xls”. 	
Resultados esperados:	Exportar los resultados en una hoja de cálculo.	
Resultados obtenidos:	Prueba Exitosa SI (X) NO ()	
Responsables de la Prueba:	Fabián Payan	
Observación:		

✖ Exportar informe

 Solo exportar archivo Excel

 Exportar y subir a Google Drive

https://drive.google.com/?tab=mo&authuser=0#my-drive

+ Fabian Andres Búsqueda Imágenes Maps Play YouTube Noticias Gmail Drive Calendar Más ▾

Google

Fabian Andres Payan Meneses 0 + Compartir

Drive

Ordenar ▾

Mi unidad

CREAR

▾ MI unidad


Compartido conmigo

Destacados


Reciente

Más ▾



<input type="checkbox"/> TÍTULO	PROPIETARIO	ÚLTIMA MODIFICACIÓN
<input type="checkbox"/> ☆ Calificaciones VA 2012.xls.xls	yo	16:19 yo



	PRUEBAS DEL SISTEMA	PS: 5
	SISTEMA INTEGRAL DE CALIFICACIÓN AUTOMATIZADA	No. 005
Fecha de Realización:	10-Noviembre-2012	
Módulo de Referencia:	Modulo Configuración General.	
Requerimientos:	<ul style="list-style-type: none"> • Permitir configurar el área a dibujar de las marcas de orientación. • Permitir configurar el área a dibujar de las zona de respuestas. • Permitir elegir el dispositivo para escanear. • Permitir configurar la cuenta de correo electrónico de Google a la cual se va a mandar los archivos y confirmación de los mismos. 	
Objetivos de la prueba:	Esta prueba se realiza para demostrar que la aplicación está en capacidad de configurar los parámetros generales a la hora de configurar y calificar el examen, como área de marcas y cuenta electrónica para cargar los archivos en la nube.	
Procedimiento:	<ul style="list-style-type: none"> • En el menú “Herramientas” de la barra principal, dar clic en “Configuración general”. • Si se va a cambiar el área de las marcas de orientación, dar clic en el cuadro de texto e indicar su ancho y/o alto. • Si se va a cambiar el área de respuestas, dar clic en el cuadro de texto de esa sección e indicar su ancho y/o alto. • Si se va a escoger un dispositivo diferente, escoger uno de la lista. Si no hay dispositivos conectados compatibles, no será posible elegir uno. • Si se va a cambiar el correo electrónico, indicar una cuenta de Google válida, con su respectiva contraseña. 	
Datos de Prueba:	<ul style="list-style-type: none"> • Área de marcas de orientación: 18px de ancho, 17px de alto. • Área de marcas de respuesta: 18px de ancho, 18px de alto. • Modelo del escáner predeterminado: Canon P-150 • Cuenta electrónica en Google: andpay05@gmail.com 	
Resultados esperados:	Almacenar los resultados en un archivo llamado “parametros.cfg”	
Resultados obtenidos:	Prueba Exitosa SI (X) NO ()	

Responsables de la Prueba:	Fabián Payan
Observación:	


 Configuración SICA

Configuración marcas predeterminadas

Tamaño marcas esquinas: Ancho(px) Alto(px)  

Tamaño cuadro respuestas: Ancho(px) Alto(px)  


Configuración dispositivo escaner

Modelo escaner predeterminado 

Configuración para la nube

Correo electrónico (para subir archivos a google)

Contraseña

 **Guardar**

Cierto Falso

- 24.- Razone cual de esas condiciones produce realimentación positiva y cual produce realimentación negativa.

26.- Ponga un ejemplo donde haya realimentación positiva, o negativa o una ley de potencias y explíquelo bien.

27.- Calcule la dimensión fractal de la alfombra de Sierpinski, que se obtiene a partir de un cuadrado, dividiéndolo en 3×3 cuadraditos, eliminando el del centro e iterando.

